Sandia National Laboratories

Exceptional service in the national interest

ME469: A Verification and Validation (V&V) Methodology (Review)

Stefan P. Domino[1,2]

[1] Computational Thermal and Fluid Mechanics, Sandia National Laboratories

[2] Institute for Computational and Mathematical Engineering, Stanford

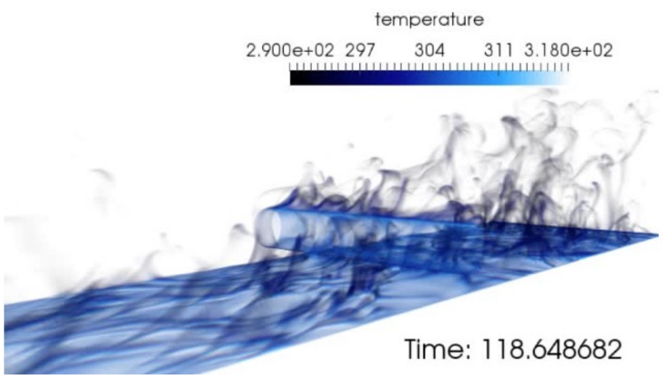
SAND2018-4536 PE

U.S. DEPARTMENT OF ENERGY

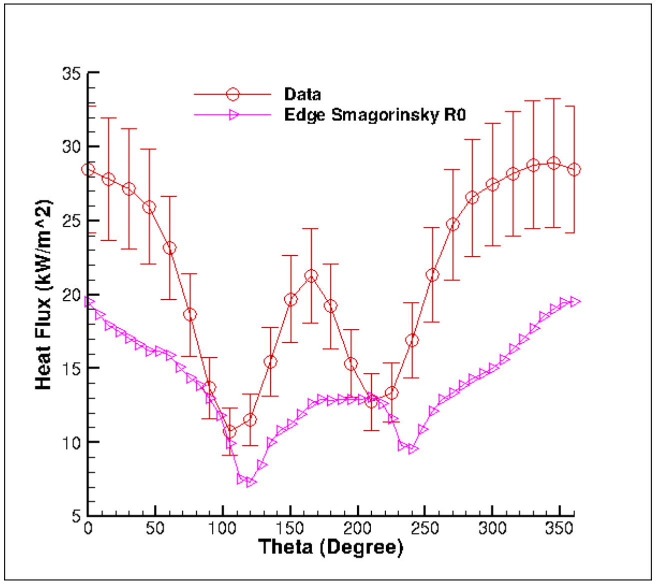
NNSA

Challenge: Understanding Errors/Uncertainties….

- One mesh, one model, unknown code/numerical pedigree…
- We need to distinguish the types of errors/uncertainties:
  - Conceptual uncertainty, δ_{input}
  - Model-form error/uncertainty, δ_{model}
  - Discretization Error, $\delta_{numerical}$
  - Code Error, $\delta_{numerical}$



Heat flux to the cylinder
Volume-rendered temperature



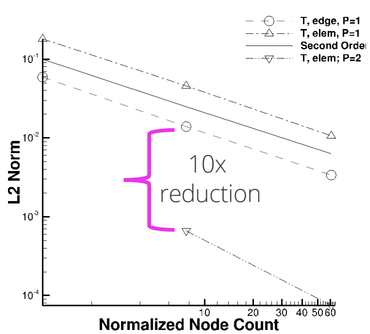
Time-averaged heat flux to cylinder

- What credible scientific hypothesis can be tested in this context?

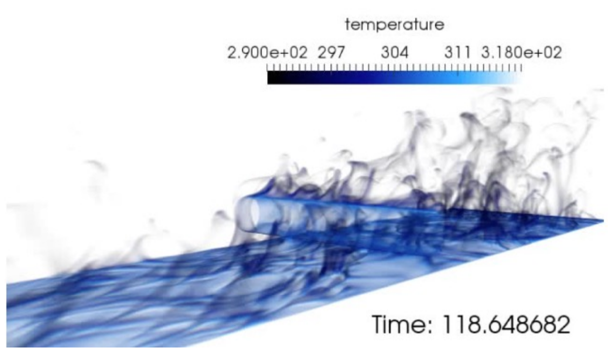
Review of a Strong V&V Process

Establish a sound LES-based V&V process (with uncertainty quantification) that includes the following attributes:

- Phenomena Identification and Ranking (PIRT)
- Code and solution verification (numerical error, $\delta_{numerical}$)
- Validation including solution sensitivity to model inputs (δ_{input})
- Structural uncertainty (model form error, δ_{model})
- Physics assumptions (your conceptual model)

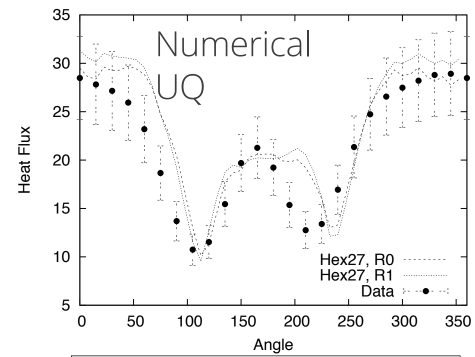


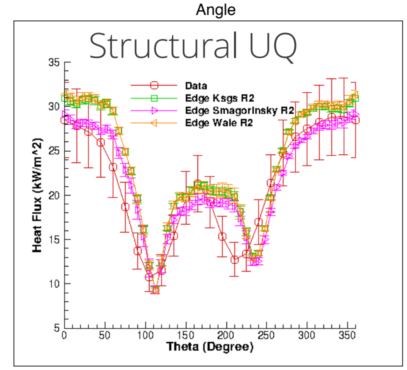
Non-isothermal MMS

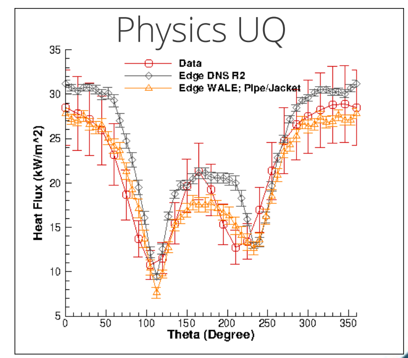


Sources of error and uncertainty in simulation

$\delta_{numerical}$, δ_{input} , δ_{model}







"An assessment of atypical mesh topologies for low-Mach LES", Domino et al., *Comp & Fluids*, 2019

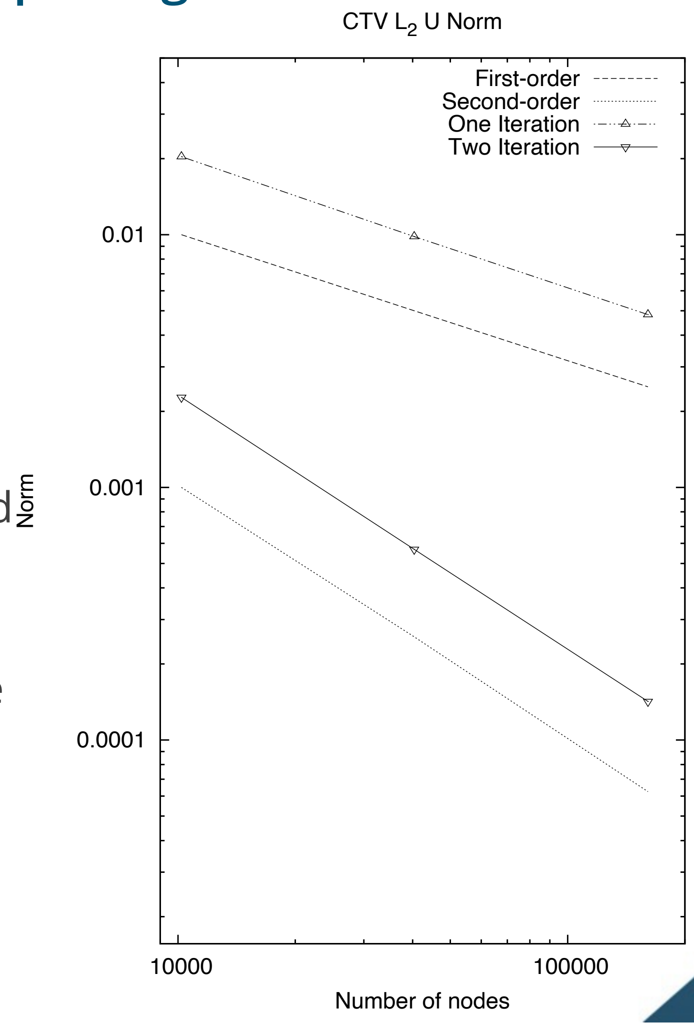
Code or Conceptual Error? Part 1: Time Splitting

Case Study: An Algorithm is thought to be second-order-in-time accurate with one nonlinear iteration: True or False?

- Issa, "Solution of the implicitly discretized fluid flow equations by operator splitting", JCP (1985).
  - Advent of the "Pressure-implicit with Splitting of Operators", or PISO
- PISO is a scheme that defines a series of predictors and correctors in the context of a fully implicit solve

Conclusion?

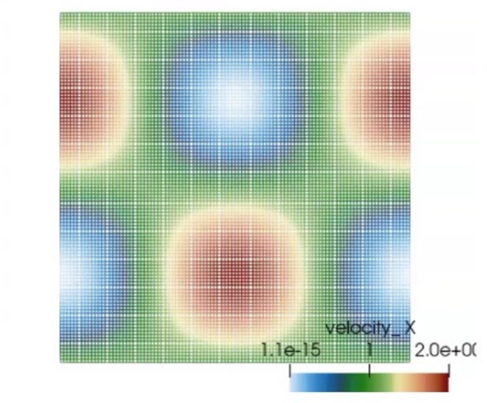
- Sometimes we code a method correctly, however, have a conceptual error in our understanding of whether or not a scheme is design-order accurate when run in the suggested manner



Consistent vs Lumped Mass Matrix

Goal: Explore the time accuracy of the consistent mass matrix approach on a structured mesh

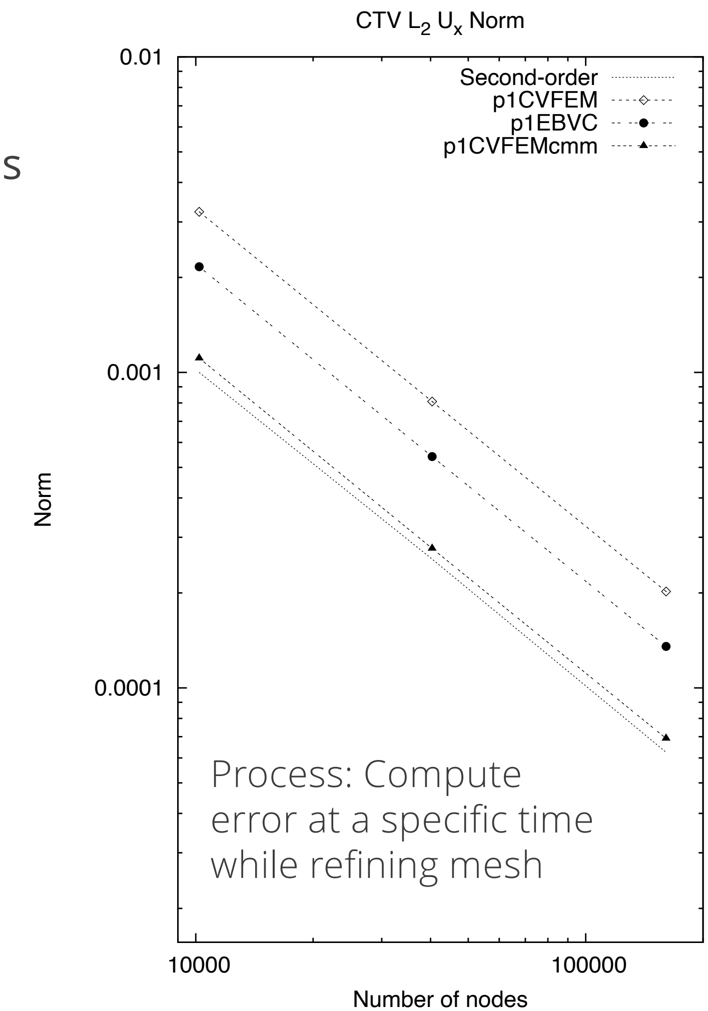
- Convecting Taylor Vortex Case Study: Analytical, transient verification problem



$$u = u_o - cos(\pi(x - u_o t))sin(\pi(y - v_o t))e^{-2.0\omega t}$$

$$v = v_o + sin(\pi(x - u_o t))cos(\pi(y - v_o t))e^{-2.0\omega t}$$

$$p = -\frac{p_o}{4}(cos(2\pi(x - u_o t)) + cos(2\pi(y - v_o t)))e^{-4\omega t}$$



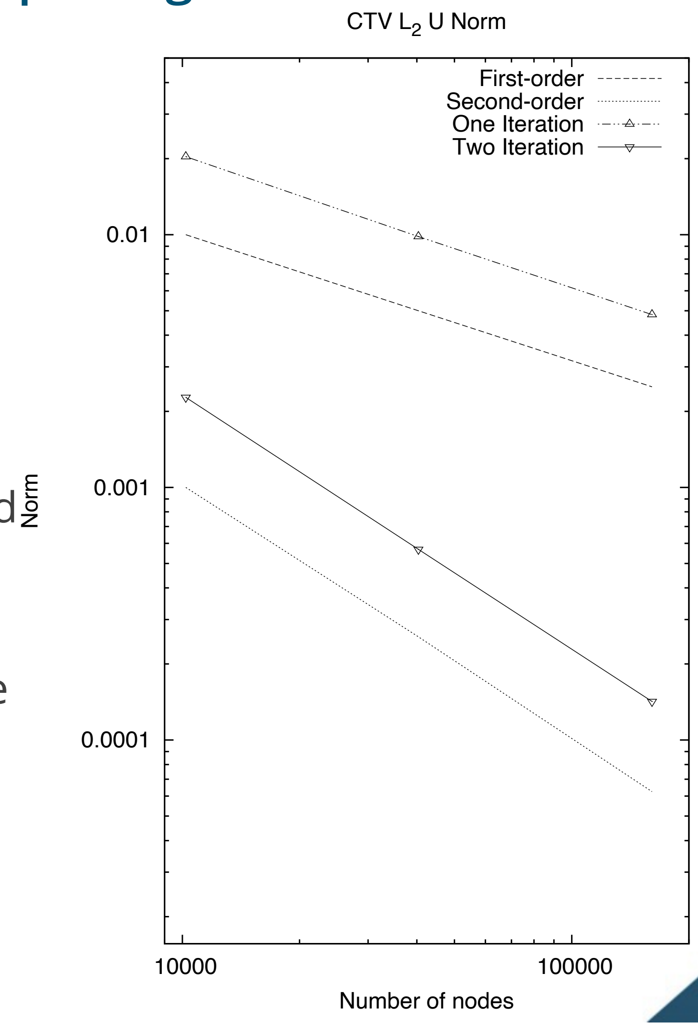
Code or Conceptual Error? Part 1: Time Splitting

Case Study: An Algorithm is thought to be second-order-in-time accurate with one nonlinear iteration: True or False?

- Issa, "Solution of the implicitly discretized fluid flow equations by operator splitting", JCP (1985).
  - Advent of the "Pressure-implicit with Splitting of Operators", or PISO
- PISO is a scheme that defines a series of predictors and correctors in the context of a fully implicit solve

Conclusion?

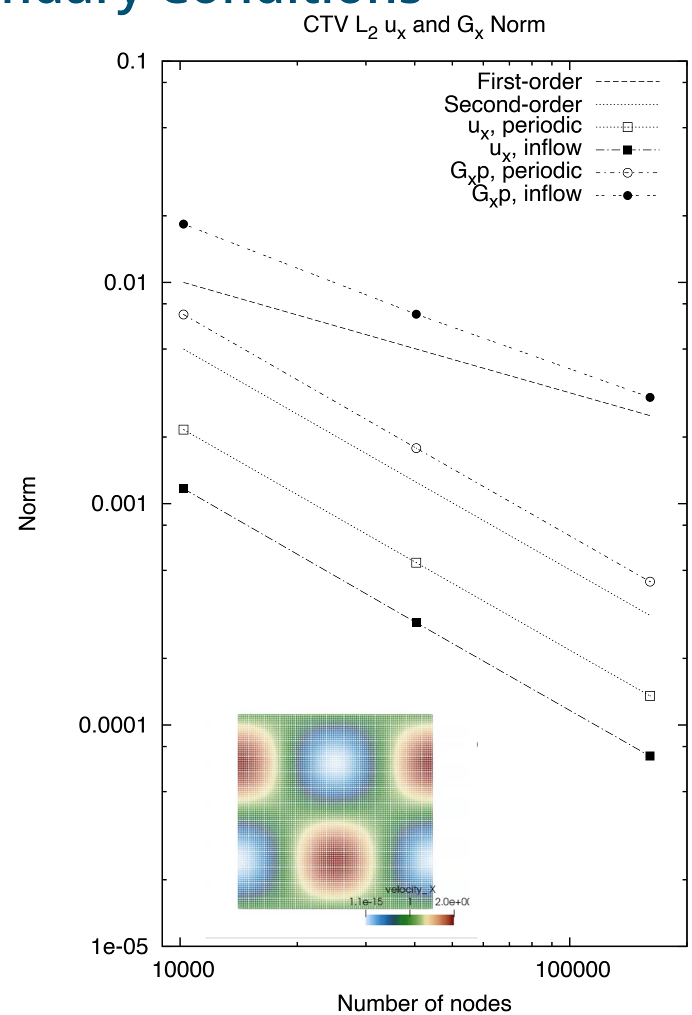
- Sometimes we code a method correctly, however, have a conceptual error in our understanding of whether or not a scheme is design-order accurate when run in the suggested manner



Code or Conceptual Error? Part 2: Boundary Conditions

Case Study: Using the convecting Taylor Vortex case, explore boundary conditions

- Choice 1: Periodic with specified initial condition
  - Requires no formal boundary conditions
- Choice 2: Inflow
  - Easy, velocity is specified as a Dirichlet
- What about pressure? The default approach is to assume a zero normal pressure gradient (no-op) – similar to what would be found at a wall
- A notion of a "spurious numerical boundary layer" (Gresho, 1995) shows a characteristic length $\delta \sim \mathrm{sqrt}(\nu\Delta t)$ error
- We also need to make sure that we are adequately converging the nonlinear system within the time step

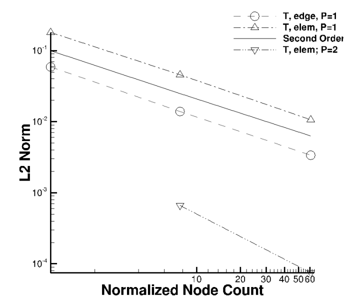


Spatial Verification Process

Process:

1. Start with a mesh of given resolution, Δx_o, or a given total node/cell-centered, etc. count, N
2. Refine the mesh, uniformly, i.e., h-refinement or in polynomial order, p-refinement
3. Run the series of refinement/promoted meshes and compute an integrated norm over the domain, L_∞, L_1, L_2, for example, below shown to be based on a nodal DOF prediction,

$$L_\infty = \sum_i max|\epsilon_i| \quad L_1 = \sum_i \frac{|\epsilon_i|}{N} \quad L_2 = \sum_i \frac{\epsilon_i^2}{N^2}$$



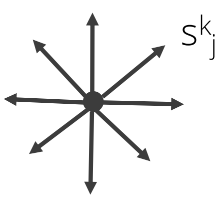
4. Evaluate error relative to what the precise solution think it should be
   - Generally easiest to observe the magnitude of reduction
   - For example, $\Delta x_1 = \frac{\Delta x_o}{2}$, for a first-order in space scheme reduces the error by 2 while for second-order, 4, and third-order, 8, etc.

Lurking, or Hidden Errors…

- In some numerical implementations, an error can exist that is not easily found at a given mesh resolution

For example:

- There may be a boundary error that manifests itself locally at a very small subset of the mesh
- There may be an error that is a driven by the fidelity of the mesh (consider our specified pressure drop in a pipe example)
- Other errors, e.g., for a discrete-ordinance method, we have an underlying *quadrature* error



$$s_j \frac{\partial I}{\partial x_j} + (\mu_a + \mu_s)\, I = \frac{\mu_a \sigma T^4}{\pi} + \frac{\mu_s G}{4\pi}$$

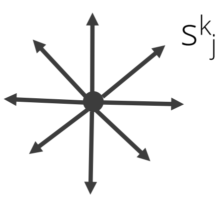
$$s_j^k \frac{\partial I^k}{\partial x_j} + (\mu_a + \mu_s)\, I^k = \frac{\mu_a \sigma T^4}{\pi} + \frac{\mu_s G}{4\pi} \qquad G \approx \sum_k w_k I^k$$

Lurking, or Hidden Errors…

- In some numerical implementations, an error can exist that is not easily found at a given mesh resolution

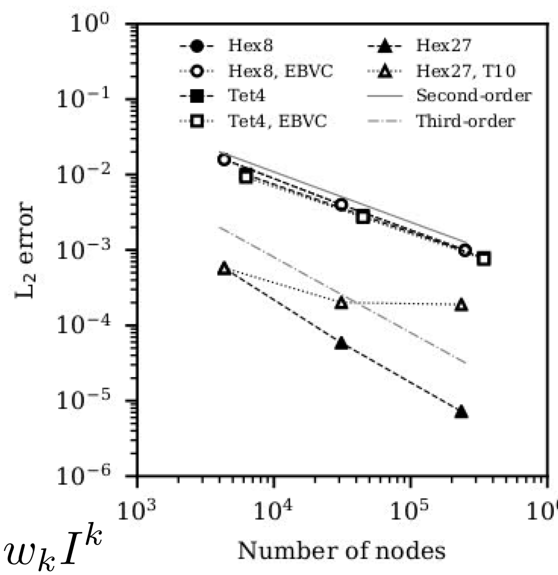
For example:

- There may be a boundary error that manifests itself locally at a very small subset of the mesh
- There may be an error that is a driven by the fidelity of the mesh (consider our specified pressure drop in a pipe example)
- Other errors, e.g., for a discrete-ordinance method, we have an underlying *quadrature* error



$$s_j \frac{\partial I}{\partial x_j} + (\mu_a + \mu_s)\, I = \frac{\mu_a \sigma T^4}{\pi} + \frac{\mu_s G}{4\pi}$$

$$s^k_j \frac{\partial I^k}{\partial x_j} + (\mu_a + \mu_s)\, I^k = \frac{\mu_a \sigma T^4}{\pi} + \frac{\mu_s G}{4\pi} \qquad G \approx \sum_k w_k I^k$$

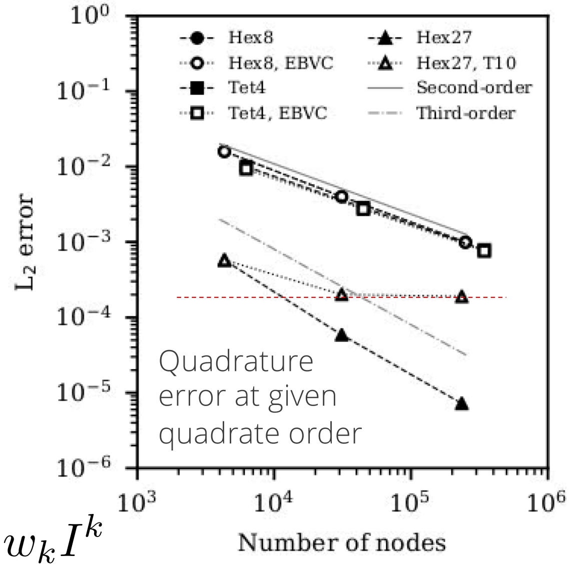


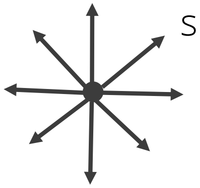
Lurking, or Hidden Errors…

- In some numerical implementations, an error can exist that is not easily found at a given mesh resolution

For example:

- There may be a boundary error that manifests itself locally at a very small subset of the mesh
- There may be an error that is a driven by the fidelity of the mesh (consider our specified pressure drop in a pipe example)
- Other errors, e.g., for a discrete-ordinance method, we have an underlying *quadrature* error



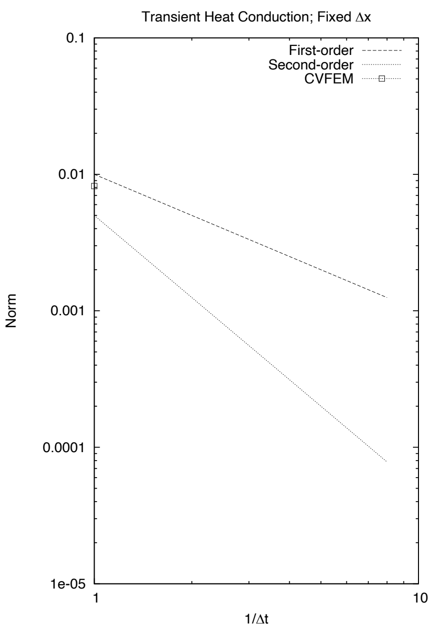


$$s_j \frac{\partial I}{\partial x_j} + (\mu_a + \mu_s)\, I = \frac{\mu_a \sigma T^4}{\pi} + \frac{\mu_s G}{4\pi}$$

$$s_j^k \frac{\partial I^k}{\partial x_j} + (\mu_a + \mu_s)\, I^k = \frac{\mu_a \sigma T^4}{\pi} + \frac{\mu_s G}{4\pi} \qquad G \approx \sum_k w_k I^k$$

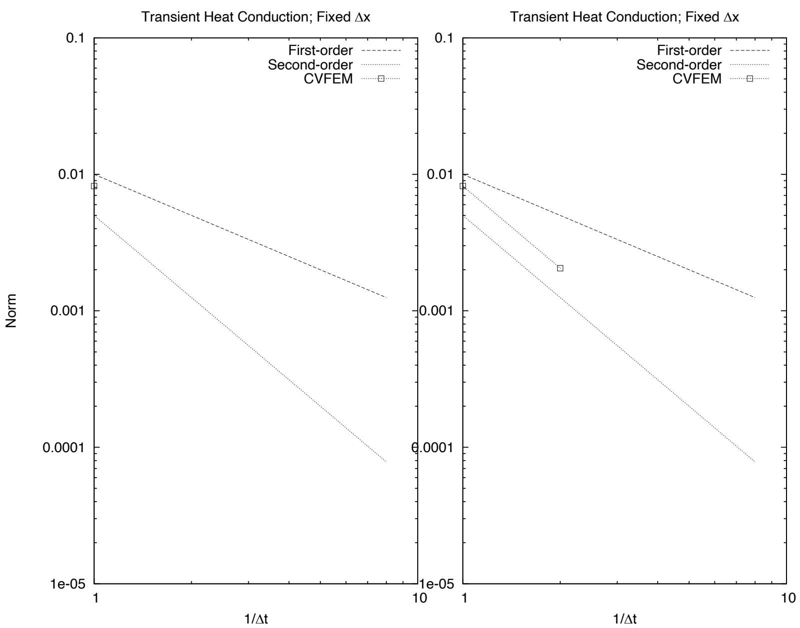
Temporal Verification: Fixed Spatial Resolution; Reduced Timestep

On a given mesh, compute error at a given time: $\epsilon \propto k_1 \Delta x^n + k_2 \Delta t^p$



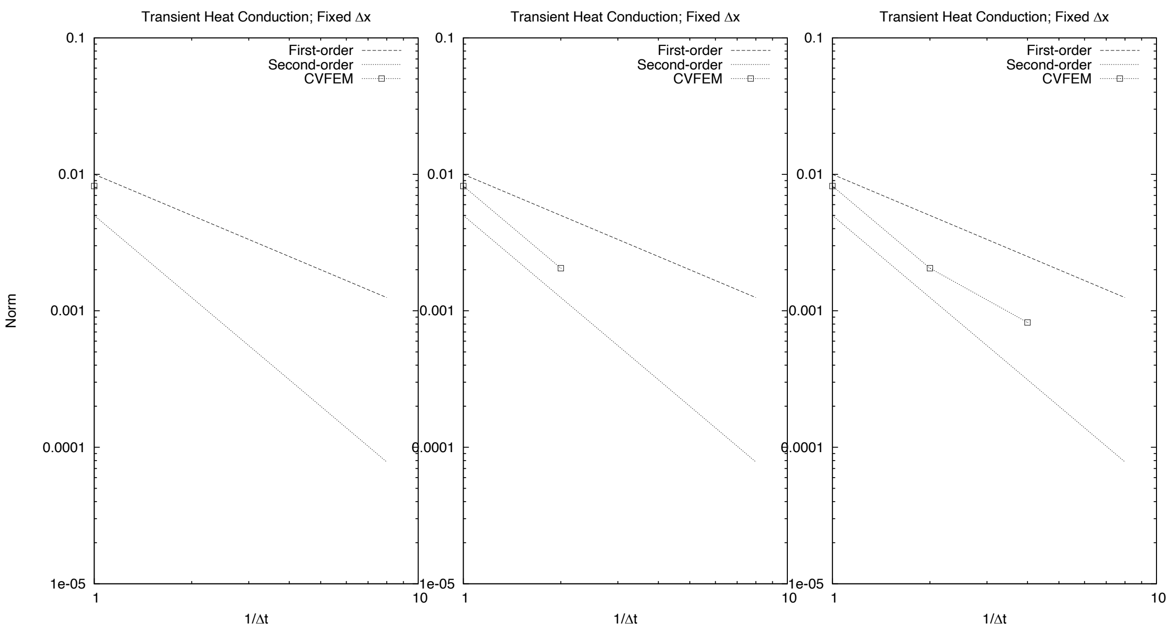
Temporal Verification: Fixed Spatial Resolution; Reduced Timestep

On a given mesh, compute error at a given time: $\epsilon \propto k_1 \Delta x^n + k_2 \Delta t^p$



Temporal Verification: Fixed Spatial Resolution; Reduced Timestep

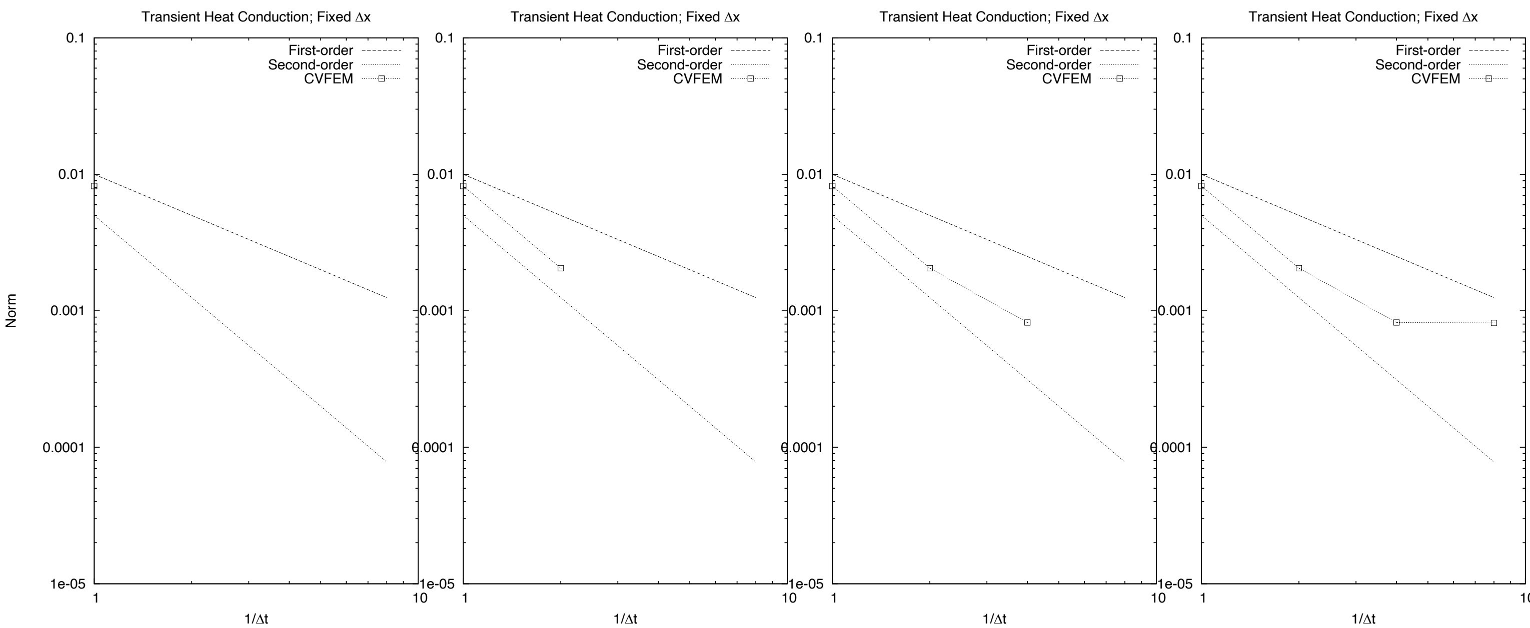
On a given mesh, compute error at a given time: $\epsilon \propto k_1 \Delta x^n + k_2 \Delta t^p$



Temporal Verification: Fixed Spatial Resolution; Reduced Timestep

On a given mesh, compute error at a given time: $\epsilon \propto k_1 \Delta x^n + k_2 \Delta t^p$

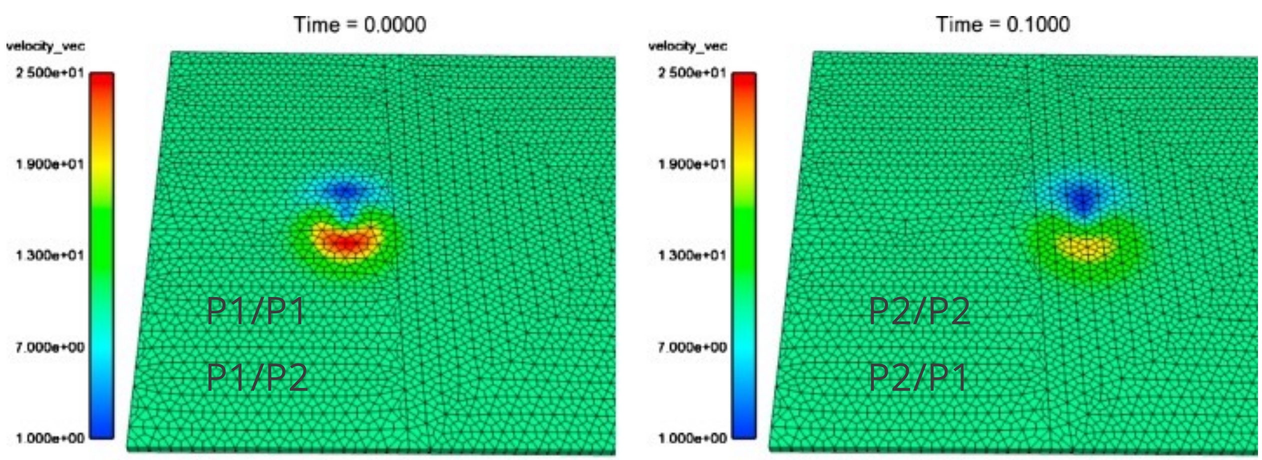
At a given mesh, the mesh spacing is fixed and, eventually, manifests itself



Temporal Verification: Reduction of Timestep as Mesh Is Refined

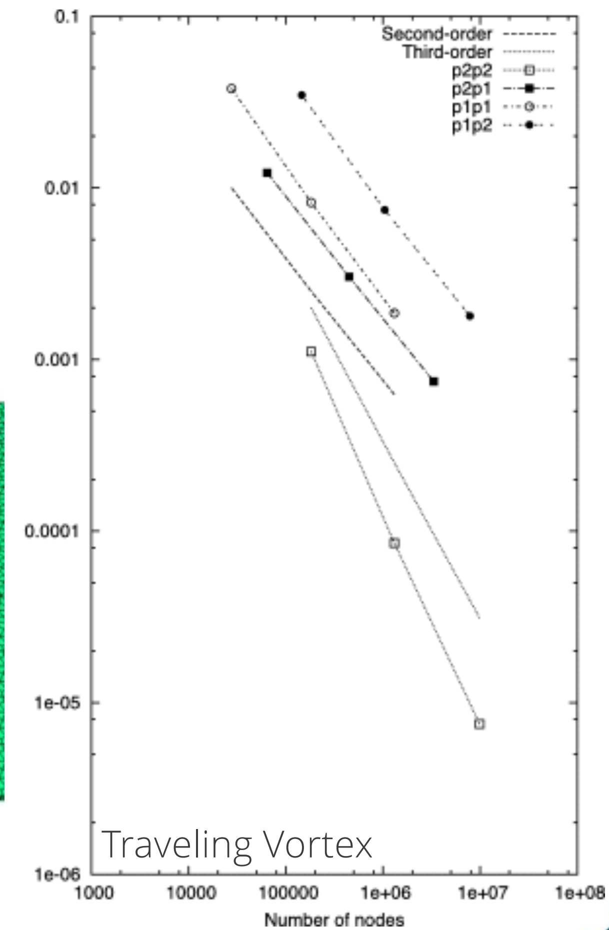
On a series of uniform mesh refinements, compute error at a given time while reducing time step

- Must ensure that the spatial error does not dominate: $\epsilon \propto k_1 \Delta x^n + k_2 \Delta t^p$



(a) Time = 0.0 seconds.

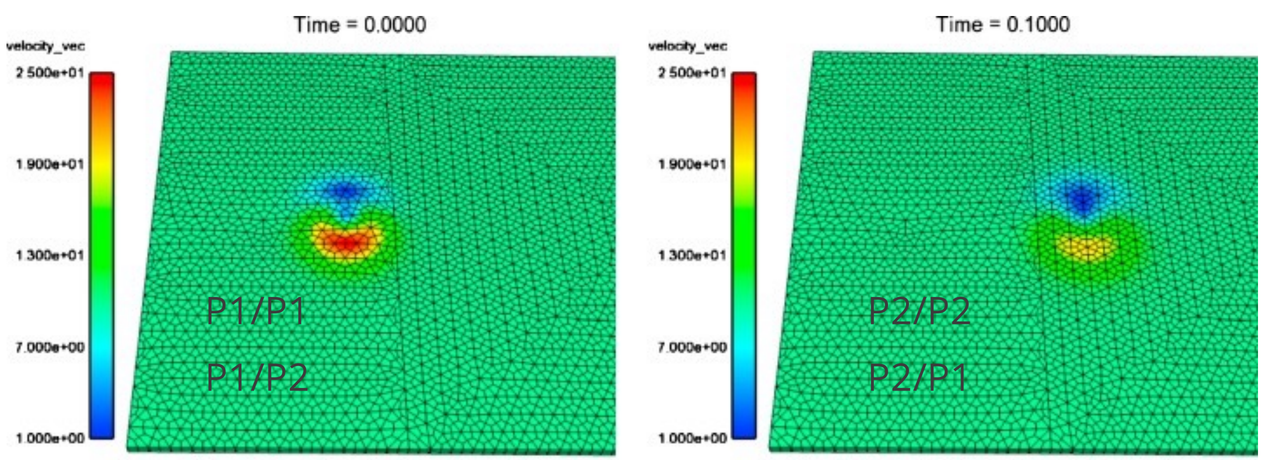
(b) Time = 0.10 seconds.



Temporal Verification: Reduction of Timestep as Mesh Is Refined

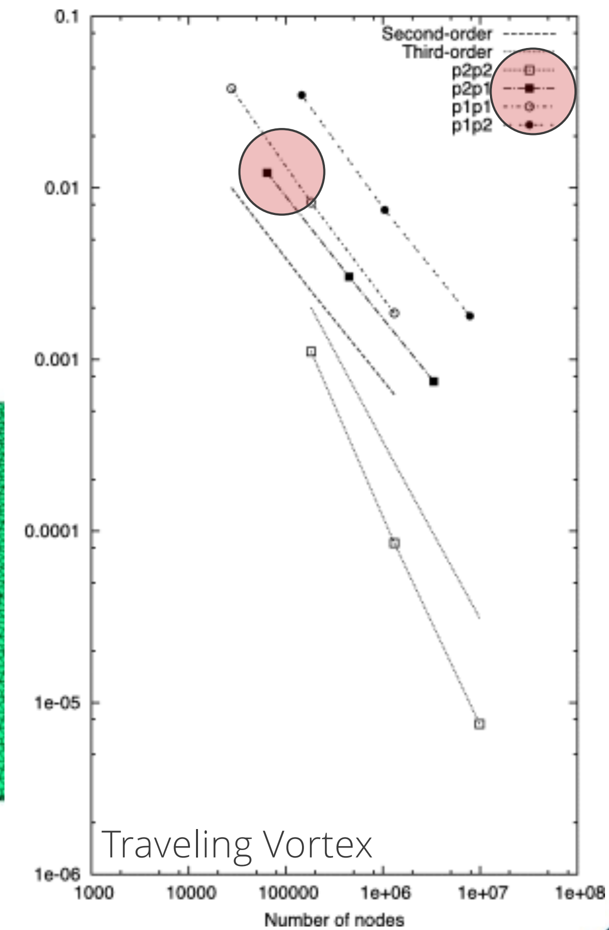
On a series of uniform mesh refinements, compute error at a given time while reducing time step

- Must ensure that the spatial error does not dominate: $\epsilon \propto k_1 \Delta x^n + k_2 \Delta t^p$



(a) Time = 0.0 seconds. (b) Time = 0.10 seconds.

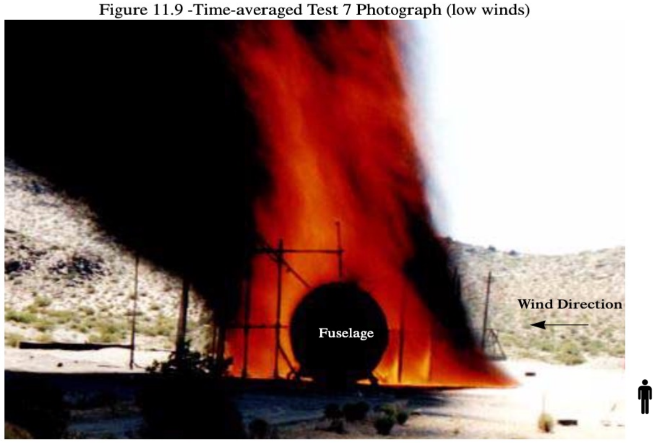
P2/P1



Domino et al., Phys. Review Fluids (2025): Fire-engulfed Object in X-wind: Can we predict ***windward→leeward*** migration of peak heat flux (and magnitude) using WMLES?

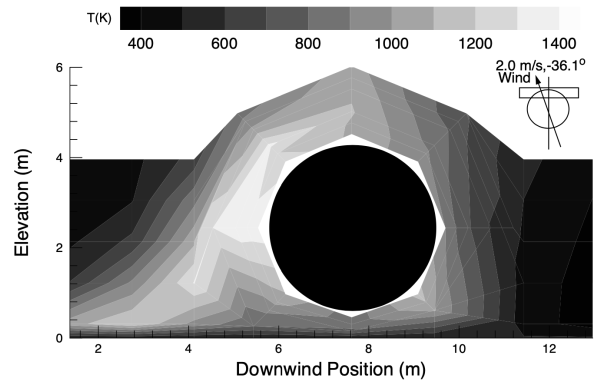
Figure 11.8 -Test 7 Photograph (low winds)

Figure 11.9 -Time-averaged Test 7 Photograph (low winds)



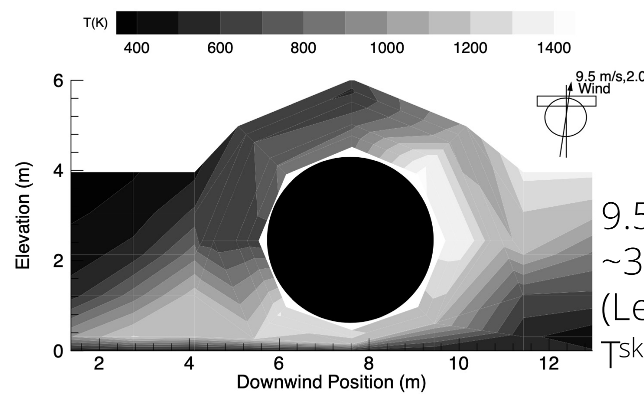
Suo Antilla and Gritzo, "The Effects of Wind on Fire Environments Containing Large Cylinders", Comb. Sci. Tech. (2008)

Wind



2.0 m/s
~220 kW/m^2
(Windward)
T^{skin} ~1400

Figure 4.35 - Test 7 Thermocouple Temperature, Windward of Center, 250-500s



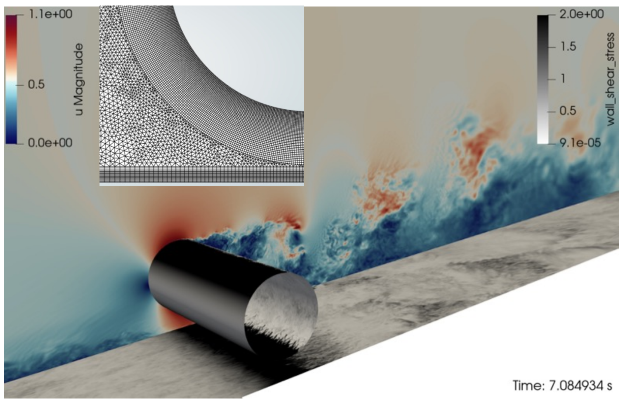
9.5 m/s
~320 kW/m^2
(Leeward)
T^{skin} ~1400

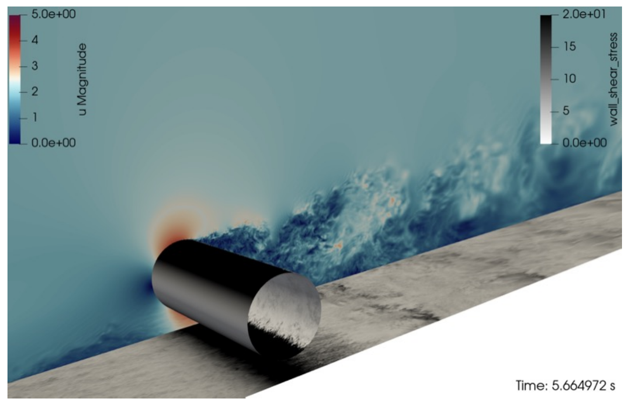
Wind

Figure 4.32 - Test 6 Thermocouple Temperature, Leeward of Center, 270-390s

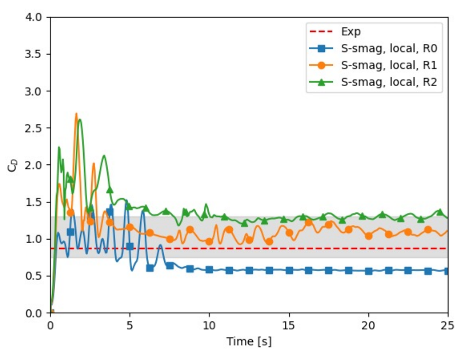
Step 1: Isothermal Crosswind Simulation Structural Uncertainty Study: Re 1.1×10^5 and 4.3×10^5 Simulations Gap (G) Diameter (D): G/D = 0.25

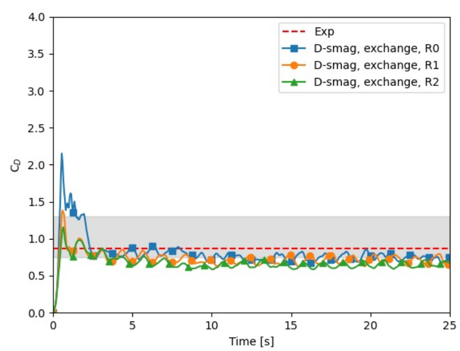
- G/D = 0.25, Yang et al., Ocean Eng. (2018) mimics the fire-engulfed fuselage use case (sub/supercritical drag regime; drag crisis well-predicted)

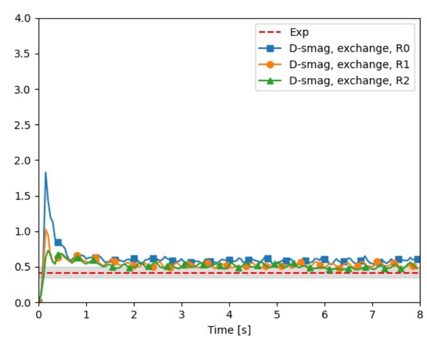


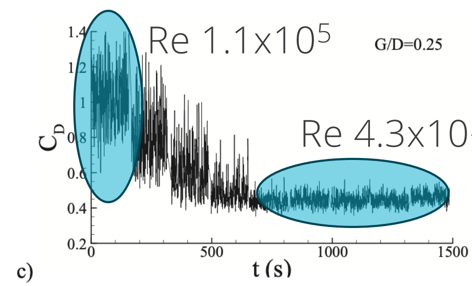


- **Finding**: Wall-modeled large-eddy simulation (WMLES) using a dynamic model with exchange-based velocity sampling increases predictivity (C_D & C_p) and convergence as a f(Δx)
- O(600) million element mesh for Refinement 2!



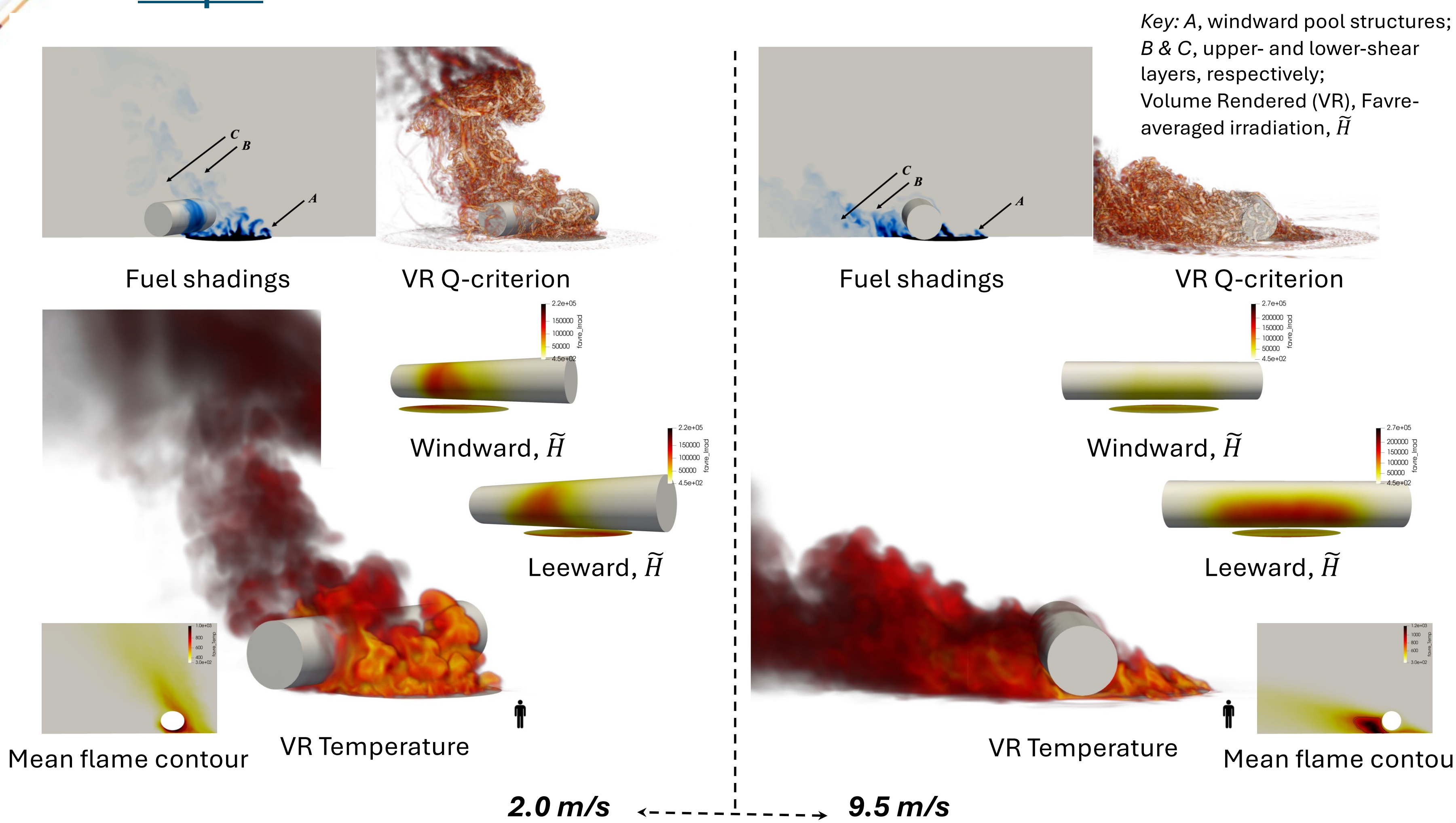






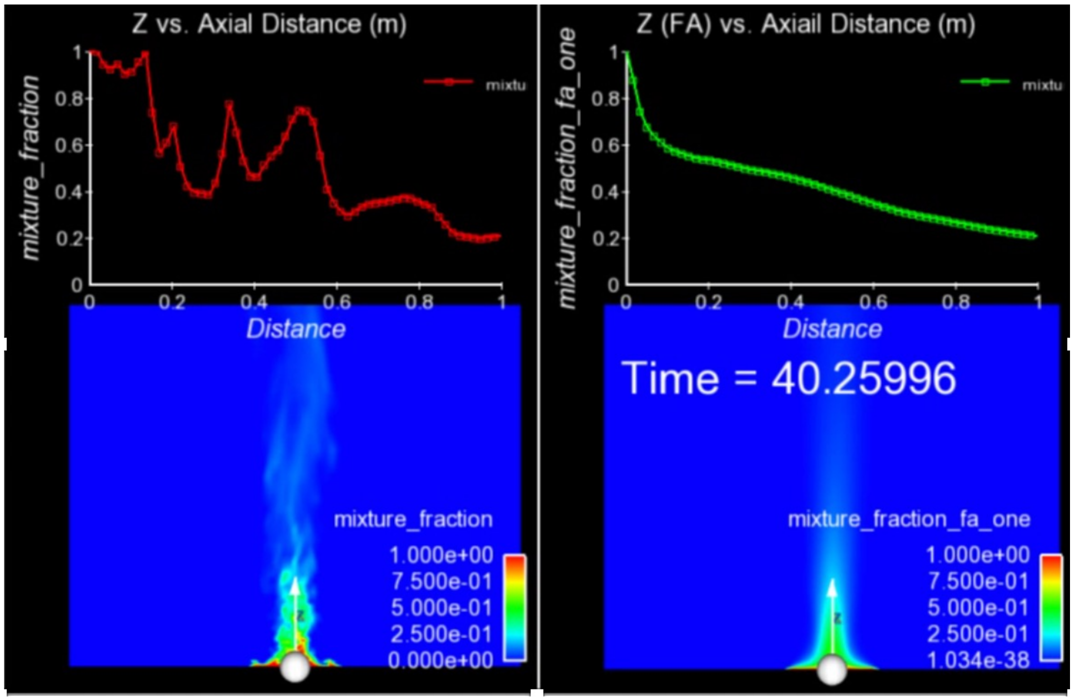
$$\mu_t = \bar{\rho} f_\mu \boxed{C_{\mu_\epsilon}} \Delta \sqrt{k_{SGS}}.$$

Step 2: Fire V&V



Finally…. For Transient Flows, Averaging is Required

- The **bane** of turbulent validation: converged statistics require many flow-through times
- Statistical convergence of a given simulation may require many flow-through-times; additional source of uncertainty and/or requirement for quantification of solution convergence



Essentials of Code Verification: Review

Taxonomy: One *verifies* code and *validates* models

- Code verification establishes the numerical accuracy of the underlying discretization for the given partial differential equation set
- Code verification seeks to provide the temporal and spatial accuracy of the underlying discretization approach

For temporal discretization error,

- A two-state Backward Euler time integrator should be first-order in time, specifically the error should scale with Δt
- A three-state BDF2 time integrator should scale with Δt^2
- A multi-stake Runge-Kutta schemes can achieve higher-order accuracy

For spatial discretization error,

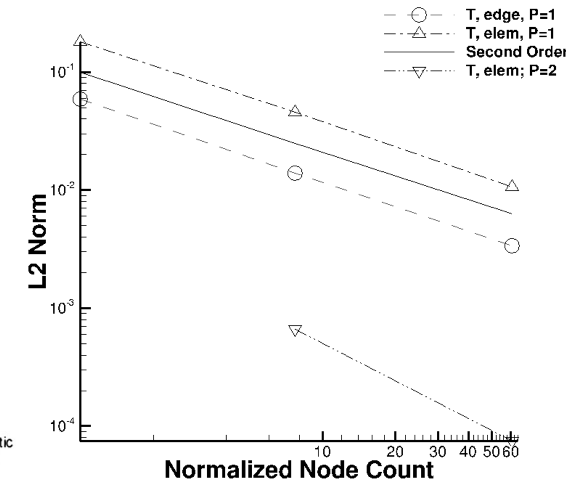
- A method is design-order if the observed order of accuracy is Δx^{P+1}, where P is the underlying basis polynomial order

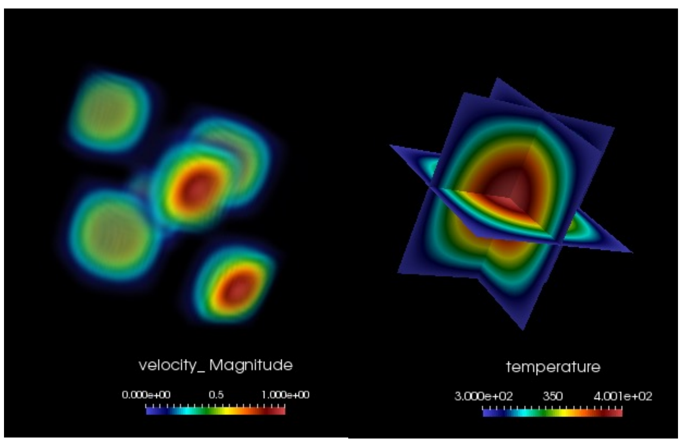
Oberkampf and Trucano, Verification and validation in computational fluid dynamics, Progress in Aerospace Sciences, Volume 38, Issue 3, 2002, https://doi.org/10.1016/S0376-0421(02)00005-2.

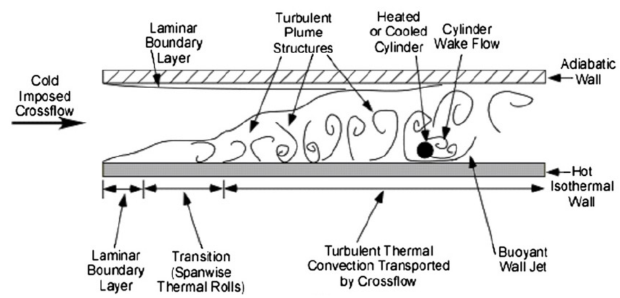
Spatial Code Verification for a low-Mach, Variable-Density Flow

	Import	Adequacy			
Convective Processes	Phen	Mod	Code	Val	Mats
Convective heat transfer	M	M	M	L	

- Density is a function of static enthalpy transport via the standard ideal gas, ρ = f(P,M,R,T)
- Temperature range maps to experiment (see below)
- Arbitrary buoyancy source term via rotated gravity vector
- Collective study now provides confidence in the interplay between numerical and modeling accuracy







Velocity Mag Temperature Kearney experimental configuration

See, "Exploring model-form uncertainties in large-eddy simulations", Domino et al, 2016

Review of the Method of Manufactured Solutions (MMS):
Providing confidence that the code implementation converges to the proper solution

- We understand that the number of analytical solutions to test our code implementation are very few in number
- How can we test the numerical accuracy of our implementation that, in general, solves very complex physics?
- Specifically, as we refine the mesh and time step, how does the error respond?

Consider a simple heat conduction PDE:

$$\rho C_p \frac{\partial T}{\partial t} - \frac{\partial}{\partial x_j} \lambda \frac{\partial T}{\partial x_j} = 0$$

With given [steady] manufactured solution:

$$T^{mms}(x,y,z) = \frac{k}{4\lambda} \left(cos(2\pi x) + cos(2\pi y) + cos(2\pi z) \right)$$

- New, analytically modified system that includes a new source term that we can implement in the code base:

$$\rho C_p \frac{\partial T^{mms}}{\partial t} - \frac{\partial}{\partial x_j} \lambda \frac{\partial T^{mms}}{\partial x_j} = S^{mms}$$

- The error is computed to be the difference between the analytical, or manufactured solution and our numerical simulation, T^h
- We can now refine the mesh and timestep, while computing the error to ensure that the rate of reduction is expected
- For example, if we believe our scheme is 2nd or 3rd order in space accuracy, one uniform refinement should reduce the error by 4x or 8x, respectively

$$S^{mms}(x,y,z) = k\pi^2 \left(cos(2\pi x) + cos(2\pi y) + cos(2\pi z) \right)$$